Implementation and Deployment

Andres Felipe Luna
40221525
MEng in Electrical and Computer Engineering
andresf.luna22@gmail.com

Sai Abhinav Tadepalli
40257238
MEng in Electrical and Computer Engineering
abhinavsai.t@gmail.com

Reynolds Duddu
40291944
MEng in Electrical and Computer Engineering
reynoldsnitrr@gmail.com

Athul Thomas
40269294
MEng in Electrical and Computer Engineering
thomasathul03@gmail.com

***Abstract*—Deliverable 3 outlines the architectural patterns used, the major design decisions taken, the advantages and drawbacks of the design decisions and the software metrics based on their granularity.**

***Index Terms*—Layered Architecture Pattern, Unit testing, Design Decisions**

I. Problem Statement

In the current era of abundant information and a vast array of books spanning various genres, it poses a significant challenge for readers to find the ideal book tailored to their preferences [1,2]. Moreover, the complexity is compounded when attempting to locate the specific availability of a book. Furthermore, the sheer quantity of books available in the market makes it even more challenging to receive personalized recommendations that align with individual reading tastes.

To tackle these multifaceted challenges, we propose to develop a book recommendation platform, named "Bibliosphere," designed to help users in their quest to discover new books. This platform not only assists users in identifying their next literary adventure but also provides real-time information on the availability and pricing of books across various e-commerce platforms using a web crawler. The bibliosphere also offers the convenience of creating user accounts thereby allowing individuals to bookmark their favorite books and provide ratings. Users also have the option to recommend a book that is not currently listed in the database, allowing them to propose its inclusion to the administrator for database incorporation. Additionally, it elevates the user experience by delivering tailored book suggestions through a sophisticated hybrid deep learning model, combining elements of content-based [6,7,8,9] and collaborative-based filtering [3,4,5].

II. Project Objective

Our project aims to develop a user-friendly and comprehensive book recommendation system that makes use of user interactions and data analysis to facilitate easy book discovery and purchase for readers with diverse reading interests. The system will utilize machine learning models and web crawlers to deliver personalized book recommendations, creating a dynamic and engaging user experience within the literary community.

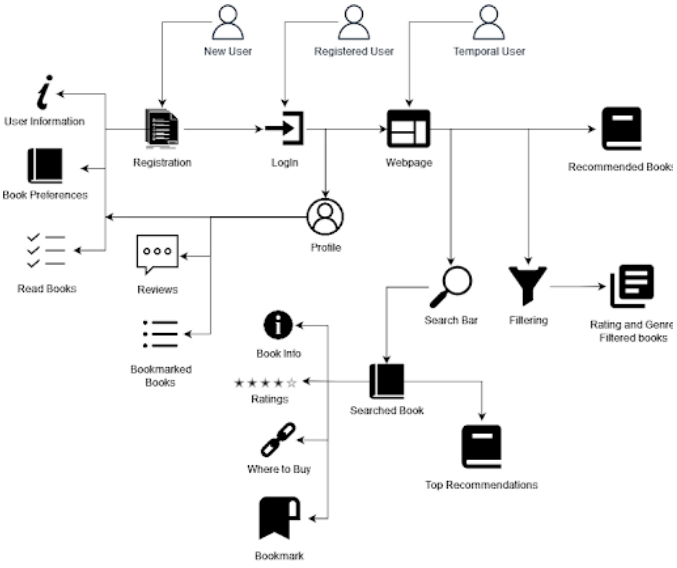


Fig. 1. System Context Diagram

III. Key Functionalities

- User Authentication and Authorization
- Personalized Recommendations based on rating and user interest
- Quick search feature
- Bookmarking feature
- Web crawler for finding purchase links
- Genre classification and categorization
- User Interaction and Engagement
- Database management

IV. Use-Cases

- User Authentication: Enable users to create accounts, log in, and manage their profiles. Implementing password encryption to protect user information.

Identify applicable funding agency here. If none, delete this.

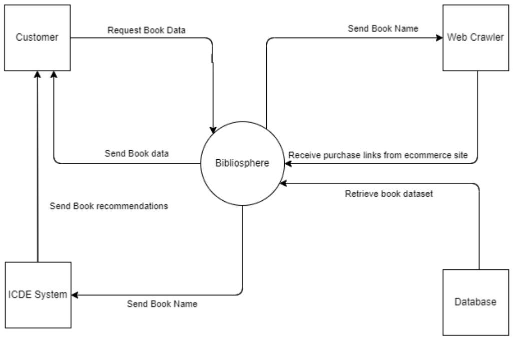


Fig. 2. Architecture of Bibliosphere

  - Precondition: The user has not registered on the system before
  - Postcondition: The user's account has been created, allowing them to log in to the system
  - Actors: User, Bibliosphere system
- Personalized Book Recommendation: The system provides personalized book recommendations based on the user's reading history and preferences.
  - Precondition: The user has provided sufficient data through the reading activity and preferences.
  - Postcondition: The system displays a list of recommended books based on the user's interest
  - Actors: User, Recommendation System
- Book Search: Users can search for specific books through the search feature
  - Precondition: The user has a book title name in mind to search
  - Postcondition: The system displays relevant search results based on the user's query
  - Actors: User, System
- Bookmark management: Users can add or remove books from their bookmark list for future reading
  - Precondition: The user is logged in and has access to book details
  - Postcondition: The selected book is added or removed from the user's bookmark list
  - Actors: User, System
- Purchase link retrieval: The system uses a web crawler to find and display purchase links for a selected book from various bookstores or e-commerce sites.
  - Precondition: The users have selected a book for which purchase links are to be retrieved
  - Postcondition: The system displays valid purchase links for the selected book
  - Actors: User, Web crawler
- Book information retrieval: Users can access detailed descriptions about a specific book, including its title, genre, user rating and reviews.
  - Precondition: The user selects a book from the search or category list.
  - Postcondition: The system provides detailed information about the selected book.
  - Actors : User, System

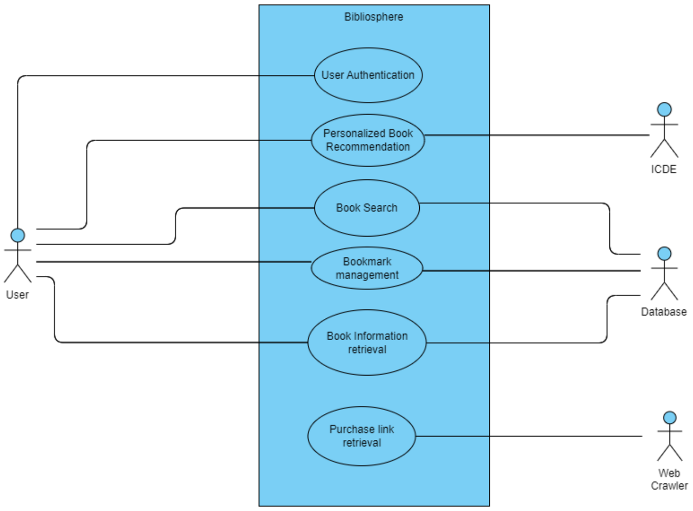


Fig. 3. Use-Case Diagram

V. User Stories

- User Registration
  - User Story Description: As a new user, I want to be able to register on the website by providing my basic information, such as name, email, and password, so that I can create a personalized account and access exclusive features.
  - Acceptance Criteria:
    - Registration Form: The registration form should include fields for the user's name, email, and password. The password field should support secure password criteria (e.g., minimum length, special characters). The form should have appropriate validation for email format.
    - Account Creation: Upon successful registration, a new user account should be created in the system.
    - Error Handling: Users should receive clear error messages for incomplete or incorrectly filled-out fields. In case of a registration failure (e.g., duplicate email), users should be notified.
    - User Access: After successful registration and email verification, users should be able to log in using their registered email and password.

- User Login
  - User Story Description: As a registered user, I want the ability to log in to the website using my email and password, allowing me to access my personalized account and view my recent activity.
  - Acceptance Criteria:
    - Login Page: The login page needs to include labeled input fields for the email and password. It should also provide a ”Forgot Password” link for users who need to reset their passwords.
    - Authentication: When users enter an incorrect email or password, they should receive an error message. If there are five consecutive failed login attempts, the user account should be temporarily locked, and the user should receive a notification.
    - Successful Login: Once successfully logged in, users should be automatically redirected to their personalized account page.
    - Security: Passwords should adhere to minimum complexity requirements, such as a specified minimum length and a combination of letters and numbers.
- Temporary User Book Discovery
  - User Story Description: As a temporary user visiting the webpage, I want to easily search for a particular book of interest or explore top-rated recommendations so that I can quickly find a compelling book to read without the need to create an account.
  - Acceptance Criteria:
    - The homepage should have a search bar for quick book searches.
    - The search functionality should provide relevant results based on book titles, authors, or keywords.
    - Users should have the option to explore top-rated book recommendations without being logged in.
  - Dependencies:
    - Availability of a book database with information for search functionality.
    - Associated rating system for generating recommendations.
- Book Search Functionality
  - User Story Description: As a user, I want to utilize the search bar to quickly find a particular book, view information about where to purchase it, and receive recommendations for similar books.
  - Acceptance Criteria:
    - The homepage should have a search bar.
    - The users should be able to type the title or keywords of a book into the search bar.
    - The search results page should display relevant books matching the search query.
    - Users should be able to click on a search result to access the detailed information about the selected book.
- Bookmarks
  - User Story Description: As a registered user, I want the ability to bookmark books that I like, so that I can easily revisit my favorite books.
  - Acceptance Criteria:
    - The user should see a ”Bookmark” button/icon next to each displayed book.
    - Clicking the ”Bookmark” button should add the book to the user’s list of bookmarked books.
    - The user should be able to view their list of bookmarked books in a dedicated section in their account profile.
    - The user should have the option to remove a bookmarked book from their list.
    - Bookmarks should persist even after the user logs out and logs back in.
  - Dependencies:
    - User authentication and account management features must be implemented.
    - The webpage must have a database to store and retrieve user’s bookmark data.
- Genre and Rating Filtering
  - User Story Description: As a user seeking personalized book recommendations, I want the ability to filter displayed recommendations based on genre and ratings, so that I can easily discover books that match my preferences.
  - Acceptance Criteria:
    - The user should see a filter option that allows selection of one or more genres.
    - The user should see a filter option that allows selection of a minimum rating.
    - Upon selecting filter criteria, the displayed book recommendations should match the selected genres and meet the specified rating.
  - Dependencies:
    - Depends on the availability of a database with genre and rating information for each book.
- Sentiment-Driven Book Recommendations
  - User Story Description: As a user with specific book preferences, I want to receive personalized book recommendations based on my reading history. Additionally, I am interested in exploring recommendations from other users with similar book preferences, so that I can discover new books that align with my preferences and with readers who share similar interests.
  - Acceptance Criteria:
    - The system should analyze the user’s reading history to generate personalized book recommendations.
    - Sentiment-related recommendations should consider positive reviews from users with similar book preferences.

    - Dependencies:
        - Requires access to the user's reading history data.
        - Depends on the implementation of sentiment analysis algorithms for book reviews.
- Book Review Feature
    - User Story Description: As a user who has read a particular book, I want the ability to share my review for the book. This will allow me to express my opinions and give insights to the community of readers interested in the same book.
    - Acceptance Criteria:
        - There should be a ”Write a Review” option visible on the book details page for users who have read the book.
        - The review form should include fields for a title, review, and a rating (e.g., on a scale of 1 to 5 stars).
        - Users should be able to submit the review form, and the system should store the review along with the user's information and the book's details.
        - Reviews should be able to be sorted by the most recent or rating value.
    - Dependencies:
        - Depends on the existence of user authentication.
- Detailed Book Information View
    - User Story Description: As a user, I want to access detailed information about a book when I select one. This includes essential details such as the title, author, and a brief overview, enabling me to quickly gather key information about the book I'm interested in.
    - Acceptance Criteria:
        - When a user selects a specific book, the system should display the book's title.
        - The author's name should be visible.
        - An overview of the book's content should be presented to the user.
    - Dependencies:
        - Availability of a database or data source containing book information.
- Ebook Availability Locator
    - User Story Description: As a user interested in a particular book, I want the ability to view a list of places where the ebook is available for purchase.
    - Acceptance Criteria:
        - The book detail view should include a section titled ”Availability.”
        - In the ”Availability” section, a list of platforms (e.g., Amazon) should be displayed.
        - Each platform in the list should have a clickable link that directs the user to the specific page where the ebook is available.
    - Dependencies:
        - Availability information may depend on the book's publication status and agreements with ebook distributors.

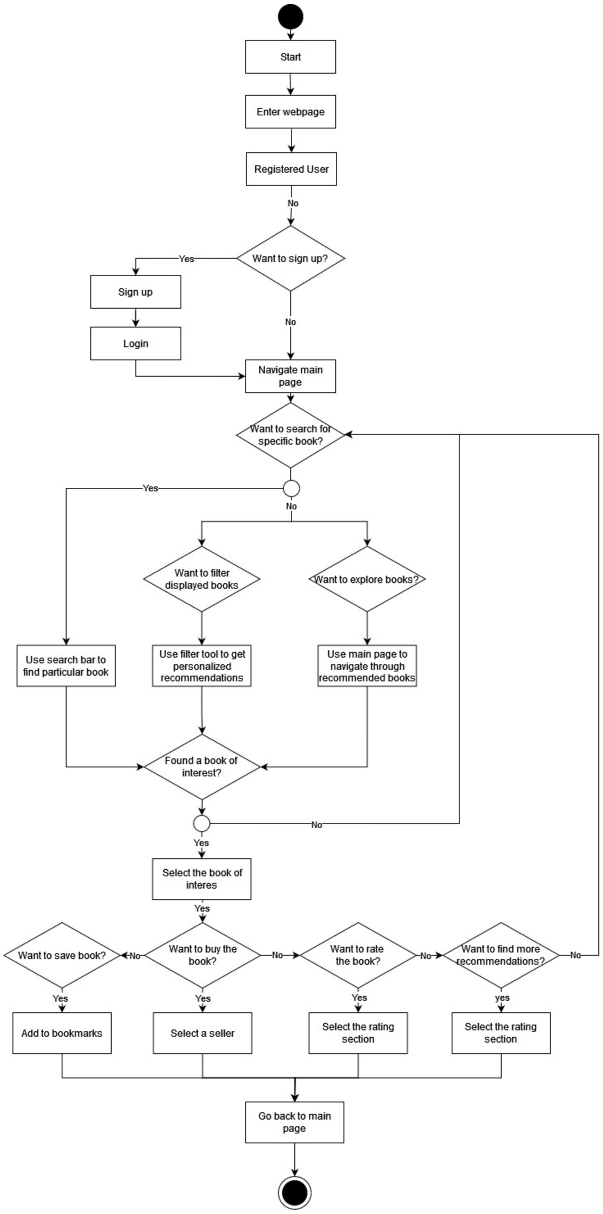

Start
Enter webpage
Registered User
No
Want to sign up?
Yes
Sign up
Login
No
Navigate main page
Want to search for specific book?
Yes
No
Want to filter displayed books
Want to explore books?
Use search bar to find particular book
Use filter tool to get personalized recommendations
Use main page to navigate through recommended books
Found a book of interest?
No
Yes
Select the book of interes
Yes
Want to save book?
No
Want to buy the book?
No
Want to rate the book?
No
Want to find more recommendations?
No
Yes
Yes
Yes
yes
Add to bookmarks
Select a seller
Select the rating section
Select the rating section
Go back to main page

Fig. 4. Activity Diagram

VI. Requirement Specification

- User Registration
    - Create the data model
    - Create a database
    - Regex for password, multiple login check
    - Check how to store user's history
- User login
    - Fetch and verify the user's credentials.
    - Lock or disable after three attempts.

  - Create a homepage layout for the user after successful login.
- Temporary user book discovery
  - Generate Random suggestions
  - Search based on genre or title and recommend book on the same.
- Book search functionality
  - Search based on present search filter and user's history.
  - Filter for search conditions according to user history or vice versa.
- Bookmarks
  - Store link to the books detailed view
  - Store the title
  - Display short preview
- Genre and Rating Filtering
  - Filter based on genre
  - Filter based on rating.
- Sentiment driven book recommendations
  - Add like/dislike feature
  - Based on like/dislike recommend appropriate books.
- Book review Feature
  - Store comments and star ratings
  - Display average ratings of the book.
- Detailed Book Information View
  - Display description/summary
  - Genres
  - Rating
  - Reviews
  - Show option to bookmark
  - Show book availability on various e-commerce sites/stores
- Ebook availability Locator
  - Search the purchase links using a webcrawler.
  - Attach the links to the current webpage

VII. Non- Functional Requirements

- Usability
  - The website layout shall allow users to reach their profile data from any page.
  - The background for all webpages will be the same.
- Accessibility
  - The website should be accessible on mobile and computer devices using a web browser when connected to internet.
- Availability
  - The website will be hosted on the Internet so that it can be accessed at all times.

VIII. Setup of Team

- Scrum: Product Owner
  - Andres Felipe Luna
- Scrum: Master
  - Athul Thomas

IX. Association of Stories

- User Story: User Registration
  - Sub Requirements: Creating Database, User profile Creation, Database update.
- User Story: User Login
  - Sub Requirements: User authentication, User validation
- User Story: Personalized Book recommendations
  - Sub Requirements: User data Analysis, Recommendation algorithm implementation
- User Story: Book Search
  - Sub Requirements: Search query processing, Search result display
- User Story: Bookmark Management
  - Sub Requirements: Bookmark creation, Bookmark Removal
- User Story: Purchase Link Retrieval
  - Sub Requirements: Web crawler setup, Link Validation
- User Story: Book Information Retrieval
  - Sub Requirements: Database query processing, Information display Interface
- User Story: Book review Feature
  - Sub Requirements: Store comments and ratings.

X. Tasks Under Each Requirement And Priority Of Tasks

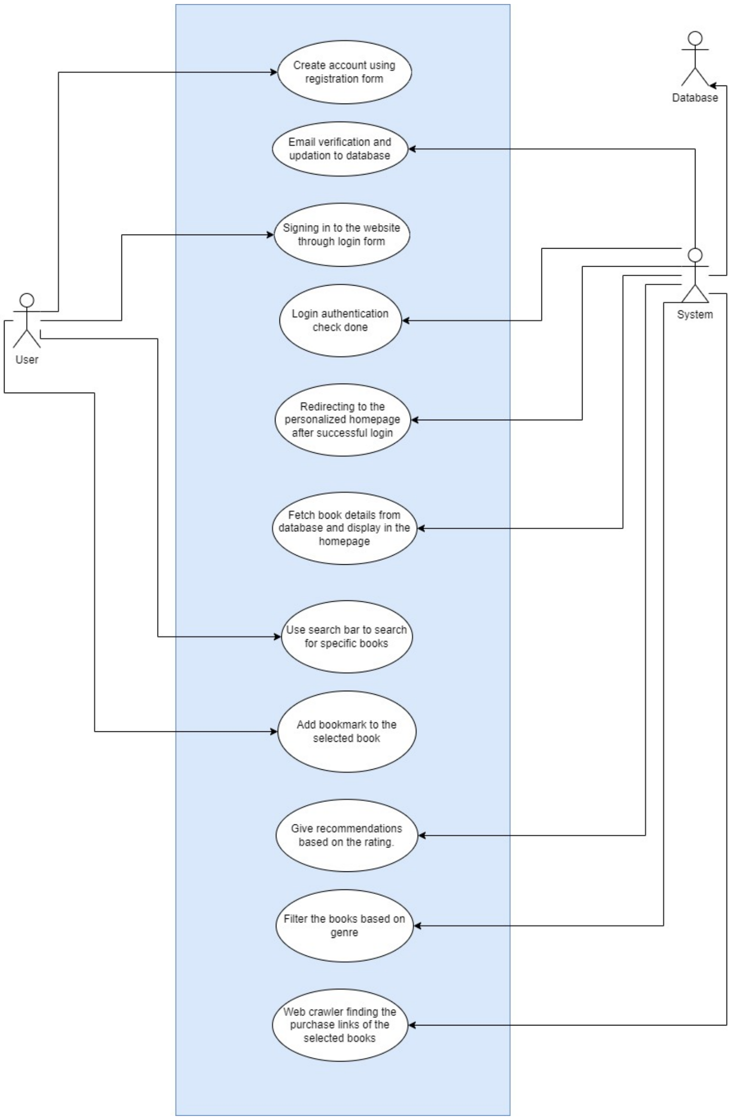
- User Registration
  - Task SP2: Create a Database
  - Task SP2: Develop user registration form UI
  - Task SP2: Update new profile in the database
- User Login
  - Task SP2: Develop user login form UI
  - Task SP2: Implement user authentication logic
- Personalized Book Recommendations
  - Task SP3: Analyse user data for preference patterns
  - Task SP3: Integrate recommendation algorithm into the system
- Book Search
  - Task SP2: Design search bar UI component
  - Task SP2: Implement search query processing backend
- Bookmark Management
  - Task SP3: Create bookmark functionality UI
  - Task SP3: Develop bookmark removal logic
- Purchase Link Retrieval
  - Task SP2: Setup web crawler framework
  - Task SP3: Validate and display retrieved links
- Book Information Retrieval
  - Task SP3: Design book information display Interface
  - Task SP2: Implement database query processing for book details.

XI. Scrum Sprint Cycle

- Sprint 2 (2 weeks):
  - Complete tasks SP2 under each requirement.
- Sprint 3 (2 weeks):
  - Complete tasks SP3 under each requirement.
- Final Demo (Sprint 4) (2 weeks):
  - Test and Debug each implemented feature.
  - Provide coverage and unit test results.
  - Document and create a presentation for demonstration.

Note : Task SP2 stands for task in Sprint 2 and SP3 for task in Sprint 3

XII. Design Diagrams



Create account using registration form
Email verification and updation to database
Signing in to the website through login form
Login authentication check done
Redirecting to the personalized homepage after successful login
Fetch book details from database and display in the homepage
Use search bar to search for specific books
Add bookmark to the selected book
Give recommendations based on the rating.
Filter the books based on genre
Web crawler finding the purchase links of the selected books
User
Database
System

Fig. 5. Use-Case Diagram (Low Level)

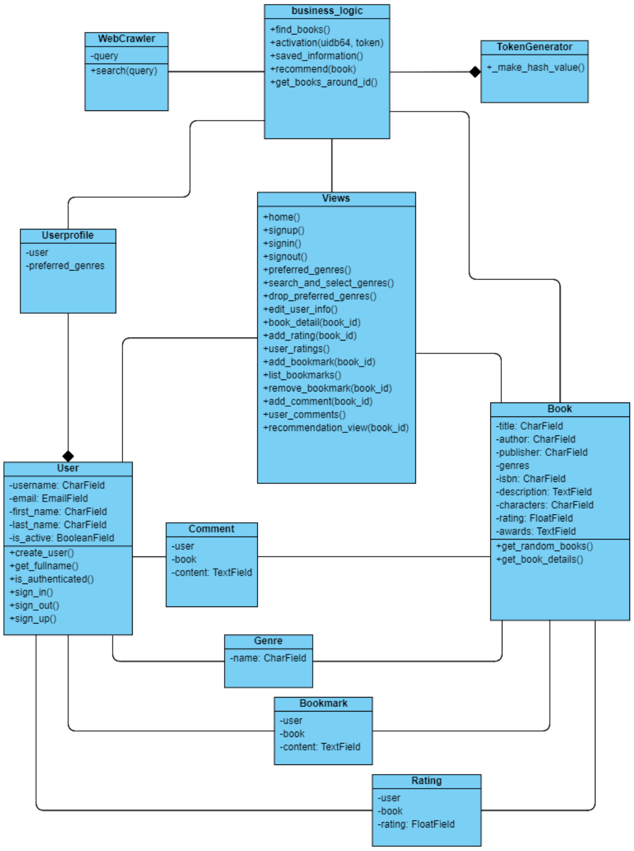


Fig. 6. Class Diagram

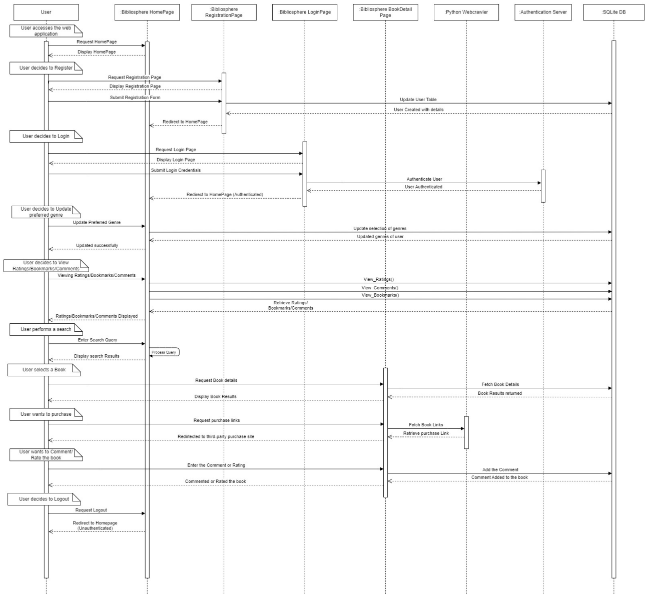


Fig. 7. Sequence Diagram

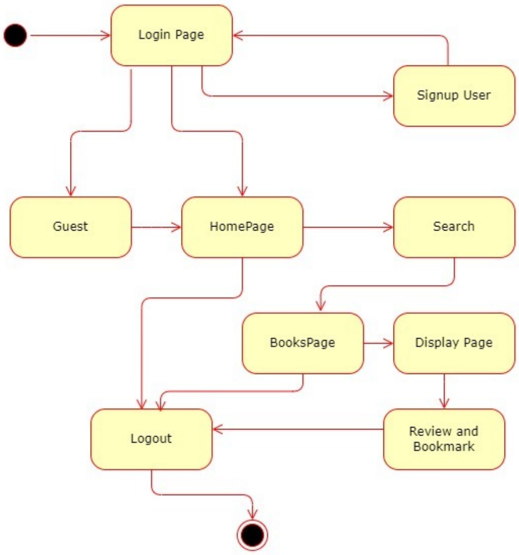


Fig. 8. State Diagram

XIII. Layered Architecture

For our project architecture, we are implementing a layered architecture as presented in the following image:

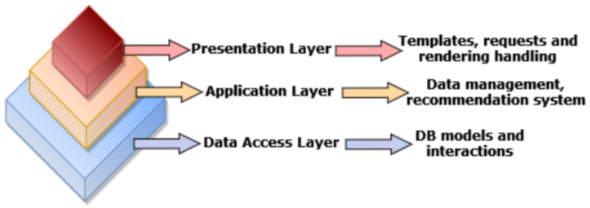

Presentation Layer
Templates, requests and rendering handling
Application Layer
Data management, recommendation system
Data Access Layer
DB models and interactions

Fig. 9. Layered Architecture

- Presentation Layer:
  - This layer includes all the HTML templates for the different sections of the website, such as the homepage, user settings, book information page, sign-in, sign-up, etc.
  - The views.py file belongs to this layer and contains functions responsible for handling HTTP requests and rendering the appropriate templates.
- Application Layer:
  - The application layer, also known as the business logic layer, defines the core logic of the book recommendation application. This includes functions for searching for books, retrieving book details, web crawling for purchase links, and providing book recommendations based on content recommendation system.
- Data Access Layer:
  - This layer includes the models for users, books, bookmarks, comments, and ratings. These models represent the data structure of the application and define how data is handled (e.g., store and retrieve).
  - It also includes database interactions such as reading and writing data to and from the database. This includes database queries and operations for retrieving user information, book data, user interactions, etc.

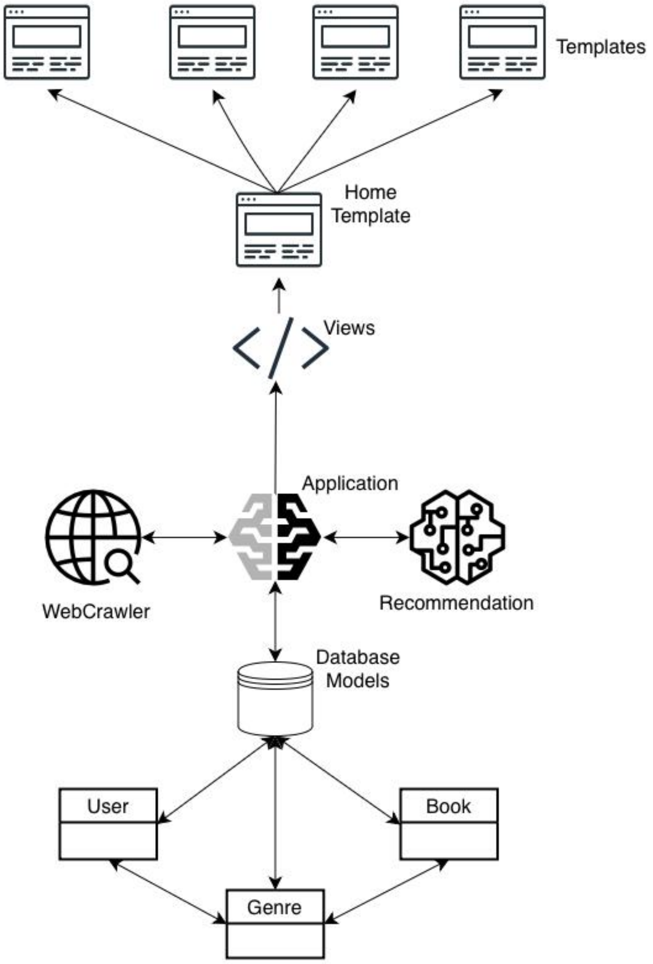


Fig. 10. Relevance of Layered Architecture to the Bibliosphere

XIV. Design Decisions

- Database Design: Using SQLite database.
- UI Design : Bootstrap framework for user friendly and interactive elements.
- Backend logic - Using Python Django framework
- User authentication and security: Django framework provides some security features
- Recommendation algorithm: Using content based filtering using cosine similarity to recommend similar books.
- Web crawler to get links: Beautifulsoup library provides a web scraping for HTML documents and returns the link for the match.
- Interface to display book details: Using HTML and Bootstrap to design the page.

- Layered Architecture pattern: Presentation, Business layer, Data Access, Database

XV. Pros and Cons of Design Decisions

- Pros-
  - SQLite is a lightweight database, making it easier to deploy and manage
  - Bootstrap ensures responsive design and visually appealing user interface.
  - Django facilitates rapid development and has built in user authentication
  - Cosine similarity is robust and can be good for content based recommendations.
  - BeautifulSoup simplifies HTML parsing without any complexity.
  - Layered Architecture facilitates modular design making it easier to manage components.
- Cons-
  - SQLite can't be used for large scale applications with high traffic.
  - Django's built in features limit customization options.
  - A single change in any layer in layered architecture can affect the entire system.

XVI. Software Metrics and Test Units

TABLE I
Software Metrics by Granularity

Task	LOC	No of Units	Component Granularity			
			Class/Object	Library	Framework	Platform
Create DB	-	-	SQLite	-	-	-
Dev. reg. form	95	1	HTML, CSS, JS	jQuery	-	-
Dev. sign-in form	65	1	HTML, CSS, JS	jQuery	-	-
Design Home UI	260	-	HTML, CSS, JS	jQuery	-	-
Design Genre UI	81	-	HTML, CSS, JS	jQuery	-	-
Analyse data	31	3	Python class	-	Django	-
Integrate recommend algo	36	1	Python class	Scikit, Pandas	-	-
Create comments UI	19	-	HTML, CSS, JS, Python class	jQuery	Bootstrap, PopperJs	-
Dev. comments logic	7	1	Python class	-	Django	-
Create ratings UI	19	-	HTML, CSS, JS, Python class	jQuery	Bootstrap, PopperJs	-
Dev. ratings logic	7	1	Python class	-	Django	-
Create bookmark UI	81	-	HTML, CSS	-	Bootstrap, PopperJs	-
Dev. bookmark logic	41	3	Python class	-	Django	-
Setup webcrawler	14	2	Python class	Beautiful Soup	-	-
Design info Interface	256	6	HTML, CSS, JS, Python class	jQuery	Bootstrap, PopperJs	-
Search Query	15	1	Python	Algolia	Django	Algolia

TABLE II
Test Units

Test Component	Test Unit	No of Testcases	Test Coverage
Models.py	test_genre_str test_user_profile_str test_book_str test_bookmark_str test_comment_str test_rating_str	12	100%
business_logic.py	test_activate test_recommend test_webcrawl_search	6	77%
Tokens.py	test_generator_token	2	100%

References

[1] G. Liden, B. Smith AND J. York. Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing, vol. 7, PP. 76-80, 2003.

[2] B. Smith and G. Linden. Two decades of recommender systems at amazon.com, IEEE Internet Computing, vol. 21, no. 3, pp. 12-18, 2017.

[3] Givon, S.,Lavrenko, V.:Predicting Social-Tags for Cold Start Book Recommendations. In: ACM Rec Sys.pp.333–336 (2009)

[4] F. O. Isinkaye, Y. O. Folajimi and B. Ojokoh. Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal, vol. 16, no. 3, pp. 261-273, 2015.

[5] S. Khusro, Z. Ali and I. Ullah. Recommender Systems: Issues, Challenges, and Research Opportunities, in Information Science and Applications (ICISA), Ho Chi Minh, 2016.

[6] F. Ricci, B. Shapira, L. Rokach and P. B. Kantor. Recommender Systems Handbook, New York Dordrecht Heidelberg London: Springer, 2011.

[7] P. Melville and V. Sindhwani. Recommender Systems, Encyclopedia of Machine Learning, 2010.

[8] A. K. Chaturverdi, F. Peleja and A. Freire. Recommender System for News Articles using Supervised Learning, 2017.

[9] R. C. Bagher, H. Hassanpour and H. Mashayekhi. User trends modeling for a content-based recommender system, Expert Systems with Applications, vol. 87, pp. 209-219, 2017.